

Design and Implementation of IoT based Smart Controlling Application for LED Scrolling Text Display with Integration of Google Assistant

Ms. Ekta¹, Dr. Vivek Kumar², Mr. Sandeep³

¹M.Tech. Scholar, Department of Electrical Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India

²Head of Department of Electrical Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India

³Assistant Professor, Department of Electrical Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India

Abstract - Here in this work efforts were made to exploit the potentials of a Voice Personal Assistant (VPA) and to enable the user to control a device remotely and quickly with much more ease. Google Assistant is an AI (Artificial Intelligence) based voice command service. In this work, using voice, we interacted with Google Assistant. It can search on the internet, schedule events, set alarms, control appliances, etc. This service we used over the smartphone. We found Google Assistant useful in remote controlling of smart home devices including lights, switches, fans and thermostats. We also found that Google Assistant supports both text or voice entry and it follows the conversation whichever entry method be used. We here in this work built an application which can control the scrolling text message over an LED dot matrix display using this Google Assistant Platform over the phone.

based Virtual Personal Assistants (VPAs) by Google i.e. "Google Assistant". The developed IoT system was NodeMCU centric and communication medium between the user mobile handset and the device to control was wireless only using the Wi-Fi protocol. The objective was to empower the user to assign some voice command over the mobile handset with VPA facility to control a smart device. The VPA should respond to the user by converting this voice command-to-text and revert back to the user by replying in voice and text both and also concurrently by performing the assigned task or transferring the commands to the device. A local hotspot was to be used to establish connectivity. Hardware prototype has validated the work.

Key Words: AI, VPA, Google Assistant, LED dot matrix display, WiFi, IoT, Cloud, ESP8266, etc.

3. SYSTEM WORKING PRINCIPLE

1. INTRODUCTION

Human's communication with the machines is evolving day by day. This human-machine interaction process started with the buttons has evolved to the touch-pads and now humans can give command to the intelligent machines by just talking to them. One of the goals of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. In recent years, the voice systems, also known as interactive conversational systems are the fastest growing area in AI. These are known as Voice Assistant or Virtual Personal Assistant (VPA) system and can be used in different areas of applications, including education assistance, medical assistance, robotics and vehicles, disabilities systems, home automation, and security access control. Nowadays, Google Assistant is becoming more and more popular personal assistant and people frequently use it's features in the applications on their, smart' devices such as smartphones, smart digital television receivers, smartwatches, etc.

This project uses two services to make it control through Google Assistant from anywhere in the world, Adafruit MQTT and IFTTT. Adafruit MQTT broker allows changing the message from any internet connected device globally. Here the NodeMCU was acting as a MQTT client hence it was constantly listening to Adafruit MQTT broker. So if any changes occur in the server side, the same changes will be observed on the client side i.e. on our NodeMCU board. And to change our message on the MQTT broker side via Google assistant, we were using one service called IFTTT. In IFTTT, we were making an applet in which we could connect two services, Google Assistant and Adafruit MQTT. So by making proper applet, we could successfully update the message on the adafruit broker side with Google Assistant on the phone.

2. AIM & OBJECTIVE

The aim of the project was to control a scrolling text message LED matrix display using Artificial Intelligence

4. SYSTEM ARCHITECTURE

Nowadays, there is technology that can retrieve contextual data from the user's environment and interact with the user in some simple, yet effective manners. The intelligent voice personal assistants can interact with the person by means of natural voice language. Here in this system a LED matrix display was controlled by the AI voice assistant system using Google Assistant. This system primarily focuses on the combination of switching circuit and the Google Assistant which is an artificial intelligence powered virtual assistant. The peerless system is distinctive from the other existing system which has been implemented

formerly. The system hardware consists of a MAX7219 driver based 32x8 LED dot matrix display module. This module was interfaced with the NodeMCU board through 5-pins. Out of these five pins, two were power pins. VCC connected to +3.3V pin of the NodeMCU board and GND was connected to the GND pin of the same board. Other three pins of the LED matrix module were control pins or data transfer pins. The MAX7219 driver communicated with the NodeMCU through SPI (Serial-Peripheral-Interface) protocol and thus only three wires were required to control the display. As the MAX7219 can control a maximum of 64 LEDs, the maximum size dot matrix display it can drive is 8x8 pixels. The 64 LEDs are driven by 16 output pins of the IC. The maximum number of LEDs that can be triggered at a time is actually eight. The LEDs are arranged in a matrix arrangement of 8x8 rows and columns. So, the MAX7219 driver activates each column periodically for a very short period of time and at the same time it also drives each row. So, by rapidly switching through these columns and rows and by exploiting the natural human eye perception limits it appeared like a continuous pattern running over the LEDs in the dot matrix display.

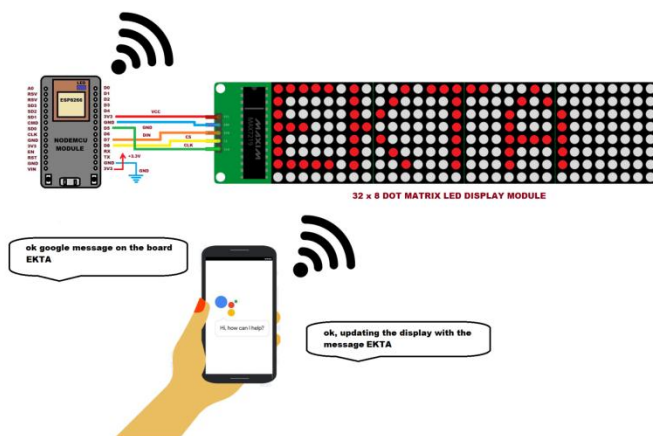


Fig.-1: System Architecture

The communication protocol used here was SPI and there is always one master device (the NodeMCU) that controls the peripheral devices or slaves (the display). We can control the display either through the NodeMCU's hardware SPI interface or three arbitrary digital pins can be configured as software SPI. Hardware SPI is faster than the software SPI. The SPI interface generally requires four pins dedicated for signals MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), SS (Slave Select).

- **MOSI** (Master Out Slave In) pin of the NodeMCU was connected to the DIN pin of the LED matrix display. This Master line sends data to the slave devices.
- **SCK** (Serial Clock) pin of the NodeMCU was connected to the CLK pin of the LED matrix

display. The clock pulses which synchronize data transmission were generated by the master.

- **SS** (Slave Select) pin of the NodeMCU was connected to the CS pin of the LED matrix display. This pin on each slave device is controlled by the master to enable and disable a particular slave device.

By using daisy chaining multiple displays, one large display can be created by simply linking DOUT (Data_Out) pin of the first dot matrix display module to the DIN (Data_In) pin of the next display module. Rest of the pins like VCC, GND, CLK, and CS are shared between all of the connected display modules. One more thing to note down here is that a common breakout board for the MAX7219 comes with a resistor between the +5V and the IC pin number 18. The resistor is used for setting the brightness or the current flow to the LEDs. The following table from the datasheet of the IC shows the value of the resistor that we should use according to the forward voltage drop of our LEDs.

5. METHODOLOGY ADOPTED

Here in this section the methodologies followed to design and implement this system using different hardware and software tools were discussed.

5.1 System Components

To build this application, these different hardware components were used

- NodeMCU – 32-bit ESP8266 development board with Wi-Fi SoC.
- MAX7219 driver based LED dot matrix display module
- +5V DC supply

Also, to build this application, three different software platforms were used

- Google Assistant
- Adafruit
- IFTTT

To use above services we need to configure these software platforms individually.

5.2 LED Dot Matrix Display

An LED dot matrix display comprised of multiple LEDs to be driven by minimum number of keys. These LEDs are arranged in a multiplexed fashion either in Cathode Column, Anode Row (CCAR) type configuration or with reversed connections Anode Column, Cathode Row (ACCR). In a dot matrix display the LEDs are placed at each of the column and row junctions of the matrix. The LEDs in the common row are connected together as are LEDs in the

common column. In the CCAR type of configuration the Anodes of the LEDs are connected to the rows and the Cathodes of these LEDs are connected to the columns. These connections get reversed in ACCR type of configuration. In matrix configuration the numbers of pins are always restricted to the number of columns and number of rows and not on the number of LEDs. For Example, a 8x8 LED dot matrix display has total 64 LEDs but can be driven by only 16 lines (8 rows and 8 columns).

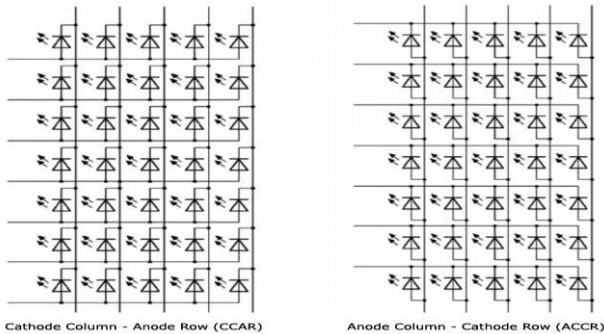


Fig.-2: Types of LED Dot Matrix Display

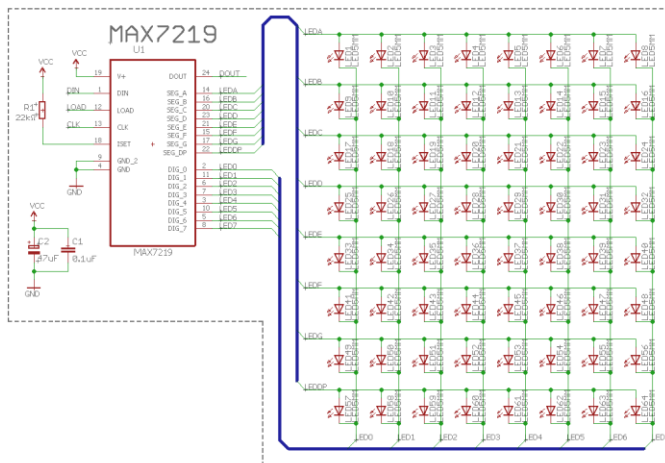


Fig.-3: Schematic Diagram of LED Display Module

5.3 SYSTEM WORK FLOW

The single layer pin-through-hole (PTH) PCB layout was designed in EAGLE. The layout editor took the reference of schematic design and generated component packages corresponding to each component chosen into the schematic design. Therefore, proper care was taken while choosing the components keeping in view of their package including parameters like type of package required (DIP, SMD, etc.) , pitch size, area, pad size, pad shape and drill hole, etc. The PCB tracks were designed in Bottom layer only. Component placement and routing is the most important aspect for a PCB design and there were other parameters too worth consideration like track size, track pitch, parasitic capacitance, track resistance, proper grounding, etc.

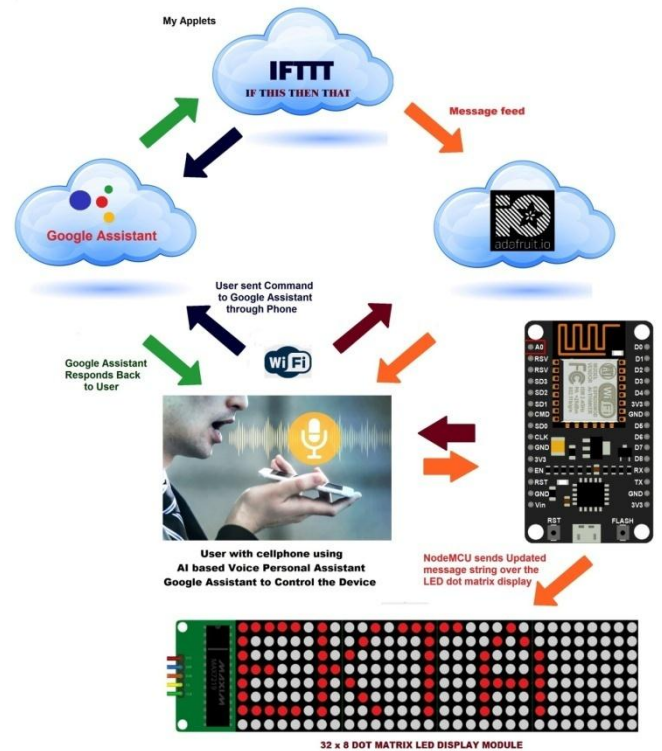


Fig.-4: System Work Flow

6. FIRMWARE

The firmware was edited and compiled in the Arduino IDE (Integrated Development Environment). The very first step before starting writing the code is to write an algorithm. So, we referred the algorithm at each and every step of firmware writing to minimize the possibility of error and minimize the time by writing an efficient source code for the system. There are certain key features of the firmware I found worth describing those below here. As an initial and necessary step the appropriate libraries were included at the beginning of firmware. The libraries used here were as shown here.

```

/***** Necessary Libraries *****/
#include <MD_MAX72xx.h>
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define SCROLL_DELAY 75
    
```

Then suitable names were assigned to the NodeMCU analog as well as digital pins with the help of #define function. The names were assigned to the pins just for the ease of user's understanding about each pin's functionality; otherwise it could be difficult to remember each pin function by its pin number. The pin definitions

as shown below were made as per the designed schematic diagram and printed circuit board (PCB). First of all download and install all the required libraries in the Arduino IDE to run the code.

```

/***** Matrix Display Pins *****/
#define CLK_PIN D5 // or SCK
#define DATA_PIN D7 // or MOSI
#define CS_PIN D8 // or SS

/***** Adafruit.io Setup *****/
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883 // use 8883 for SSL
#define AIO_USERNAME "VIKRANT_IOT"
#define AIO_KEY "aio_DbNi35gBVsydyWFpPVRgBPjwEptA"

```

In the source code, first of all we need to enter the ssid and password of our router or mobile hotspot. You can add multiple ssid names and passwords so that if any one of the given routers is available it will automatically get connected to that router. You need to provide your Adafruit account credentials. For that just visit the website with domain name io.adafruit.com. Go inside the dashboard we created. Click on this Key icon and copy the username and active key and paste it into your code. So, when I use Google Assistant on my mobile and give voice command as "Ok Google, message on the board Ekta", applet created in IFTTT receive this command and will send text data 'Ekta' to the Adafruit feed. This will trigger the event on Adafruit dashboard which is continuously monitored by the microcontroller (here NodeMCU). This microcontroller will take action as per the data change on the Adafruit dashboard.

```

WiFi.mode(WIFI_STA);
WiFiMulti.addAP("Ekta", "pass321");
WiFiMulti.addAP("Redmi", "abcd#1234");
WiFiMulti.addAP("Ekta2", "abcd#4321");
Serial.println("Connecting");

```

7. EXPERIMENTAL SETUP & RESULTS

The 32x8 LED dot matrix display module was interfaced to the NodeMCU board. The whole system was powered up by a +5V regulated dc power supply. Other DC power sources like battery, adaptors, SMPS could also be used. The current consumption of the system was very low and thus it could be powered even by a laptop's USB port. As already discussed in previous sections it could be observed that how multiple MAX7219 driver ICs could be daisy chained to drive multiple LED dot matrix modules. Here an LED dot matrix display device was controlled by an AI powered Voice Personal Assistant i.e. Google Assistant. The Google Assistant was downloaded and installed in the user's mobile phone. An Adafruit account was also created to create an application specific dashboard in that platform with some other credentials. Another open source cloud platform IFTTT (If This Then That) was installed into the user's handset. There were simple steps to get these softwares installed. Then to

establish connection between the hardware system and these software platforms and the Google Assistant, some user credentials like mobile hotspot ssid, password and Adafruit username and active key were entered in the source code. The source code was compiled and uploaded into the NodeMCU and powered up. The mobile hotspot in the user's cellphone was turned ON. The system automatically established the wireless connection through Wi-Fi hotspot and the LED dot matrix display started with default message string scrolling over it continuously. User turned on the Google Assistant in his/ her phone and gave voice commands to the Google Assistant. The Google Assistant immediately within few seconds responds back by voice only accepting the command given by the user and displayed the text on LED dot matrix display. The work was tested, calibrated and validated after multiple iterations.



Fig-5: MAX7219 Driver Based LED Dot Matrix Display

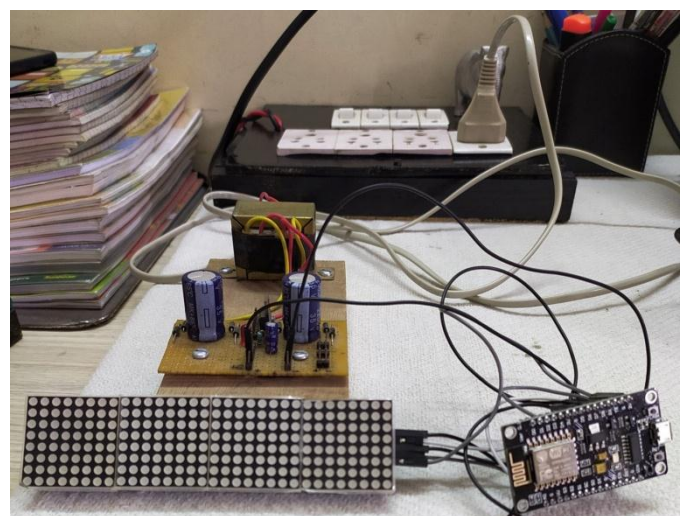


Fig -6: Experimental Setup

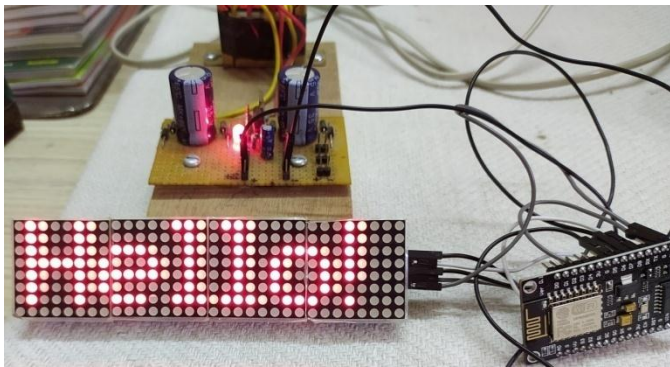


Fig.-7: 'Hello!' Text Scrolling over LED Dot Matrix Display

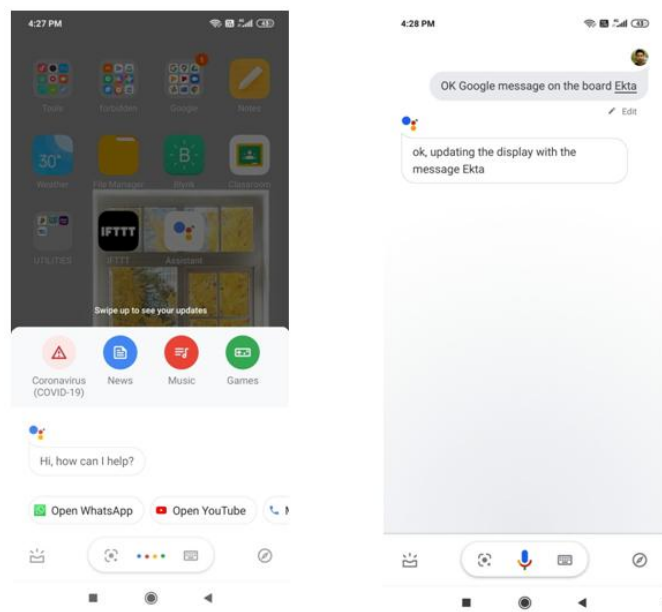


Fig.-8: User Giving Voice Command to the Google Assistant

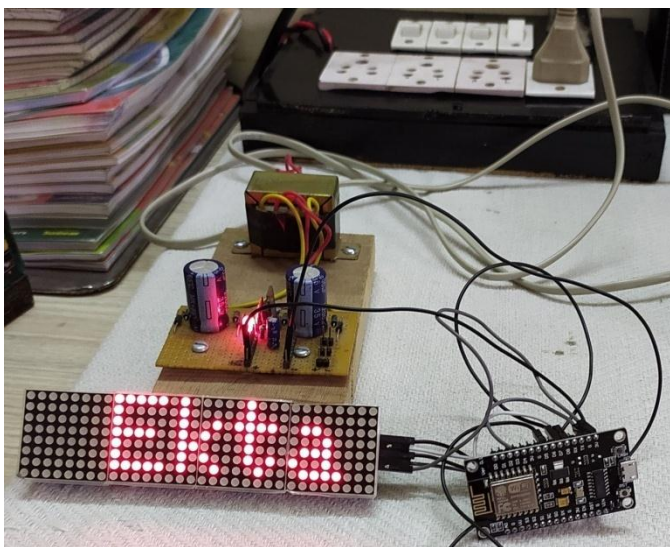


Fig -9: Updated Text String Displayed on LED Display

Here as shown in the figure a text string of my name 'Ekta' got displayed by me for the demonstration purpose only. Many text strings of varied length were tried and tested over this platform. Also both the Voice Commands as well as Typing Text commands were tested here on the Google platform. So, the hardware system developed here validated the work.

8. CONCLUSION DRAWN

Finally an application has been implemented which included technologies like Artificial Intelligence based Voice Personal Assistant (VPA), transferring data over the Cloud, IoT based hardware system using Wi-Fi protocol for establishing communication. In short it was an IoT based application that utilizes another AI powered platform for controlling. So Google Assistant controlled LED scrolling message display was successfully implemented here in this work. As it was an AI powered system it was completely user friendly and could accept commands from different users. This system provided user the liberty to use either a voice based command or a text command. The developed system was highly responsive and performance wise much more efficient than the conventional systems available. Updating a message on the display has become very quick now by adopting this technique. Overall it's a worthy system.

9. FUTURE SCOPE

This work can be further extended to new levels as adoption rate of AI is increasing at a rapid pace and IoT has already captured the market. The combination of two would lead to the development and implementation of much more sophisticated systems. These systems would restrict human interventions as most of the tasks would be effectively and efficiently performed by these smart systems only. Google Assistant based controlling of different parameters and devices could be quite useful for disabled patients or elder people. One can integrate multiple sensors and home appliances with the help of IoT platforms and their controlling could be made easy using these AI powered VPAs like Google Assistant over the phone.

REFERENCES

- [1] Haris Isyanto; AjibSetyo Arifin; Muhammad Suryanegara, "Design and Implementation of IoT-Based Smart Home Voice Commands for disabled people using Google Assistant", International Conference on Smart Technology and Applications (ICoSTA), 2020, Publisher: IEEE
- [2] Tae-Kook Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant", International Conference on Electronics,

- Information, and Communication (ICEIC), 2020, Publisher: IEEE
- [3] Vanathi K. Lalitha; B. Mahalakshmi; S. Madhusudan; M. Srinivasaperumal; S. Srikanth; Sathish R. Kumar, "Smart Control Of Home Amenities Using Google Assistant And Clap Switch Circuit", 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, Publisher: IEEE
- [4] David Sheppard; Nick Felker; John Schmalzel, "Development of Voice Commands in Digital Signage for Improved Indoor Navigation Using Google Assistant SDK", IEEE Sensors Applications Symposium (SAS), 2019, Publisher: IEEE
- [5] Mokh. SholihulHadi; Maulana Ahmad As Shidiqi; Ilham Ari ElbaithZaeni; Muhammad Alfian Mizar; Mhd Irvan, "Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant", International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2019, Volume-6, Publisher: IEEE
- [6] İlkan Yıldırım; Erkan Bostancı; Mehmet Serdar Güzel, "Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and GoogleAssistant Build-In Smart Home Speakers", 4th International Conference on Computer Science and Engineering (UBMK), 2019, Publisher: IEEE
- [7] M. Sundarramurthi; A. M. Anjana Sundari; Anandi Giridharan, "A Method of Designing Home Automation Control System (HACS) Using Virtual Assistant And Mobile Application", International Conference on contemporary Computing and Informatics (IC3I), 2019, Publisher: IEEE
- [8] Veton Këpuska; Gamal Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)", IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, Publisher: IEEE
- [9] Septimiu Mischie; Liliana Mățiu-Iovan; Gabriel Gășpăresc, "Implementation of Google Assistant on Raspberry Pi," International Symposium on Electronics and Telecommunications (ISETC), 2018, Publisher: IEEE
- [10] Aleksandar Lazić; Milan Z. Bjelica; Dejan Nad; Branislav M. Todorović, "Google Assistant Integration in TV Application for Android OS", 26th Telecommunications Forum (TELFOR), 2018, Publisher: IEEE
- [11] UrošVišekruna; Milan Savić, "Integration of Google Assistant in Android Application for Voice Control of Media Playback", 26th Telecommunications Forum (TELFOR), 2018, Publisher: IEEE
- [12] Arsénio Reis; Dennis Paulino; Hugo Paredes; Isabel Barroso; Maria João Monteiro; Vitor Rodrigues; João Barroso, "Using intelligent personal assistants to assist the elderlies an evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri", 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW), 2018, Publisher: IEEE
- [13] Sandipan Chakraborty; Shouvik Mukherjee; Suvrojit Kumar Saha; Himadri Nath Saha, "Autonomous Vehicle for Industrial Supervision Based on Google Assistant Services & IoT Analytics", IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2019, Publisher: IEEE
- [14] Laura Burbach; Patrick Halbach; Nils Plettenberg; Johannes Nakayama; Martina Ziefle; André Calero Valdez, ""Hey, Siri", "Ok, Google", "Alexa". Acceptance-Relevant Factors of Virtual Voice-Assistants", IEEE International Professional Communication Conference (ProComm), 2019, Publisher: IEEE